

[illegible]

(2)	167	Declarations
(3)	187	TTY\$PUTNEXTCHAR - BUFFER CHARACTER
(4)	260	BUFFER_CHAR - puts character into typeahead buffer
(5)	357	BUFFER_INSERT - INSERT CHARACTERS INTO THE TYPEAHEAD BUFFER
(6)	404	Put next character service routines
(7)	511	PRE-TYPEAHEAD CONTROL CHARACTER ACTION ROUTINES.
(9)	520	CONTROL C, CONTROL Y handlers
(10)	598	CONTROL O handler
(11)	652	CONTROL Q handler
(12)	682	CONTROL S handler
(13)	711	CONTROL X handler
(14)	740	ESCAPE, CSI handler
(15)	756	End of module


```
0000 1      .TITLE  TTYCHARI - terminal input character routines
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10
0000 11      *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12      *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13      *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14      *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15      *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16      *  TRANSFERRED.
0000 17
0000 18      *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19      *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20      *  CORPORATION.
0000 21
0000 22      *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23      *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28 ++
0000 29
0000 30 FACILITY:
0000 31
0000 32      VAX/VMS TERMINAL DRIVER
0000 33
0000 34 ABSTRACT:
0000 35
0000 36      THIS MODULE CONTAINS ROUTINES NEEDED FOR CHARACTER INPUT.
0000 37
0000 38 AUTHOR:
0000 39
0000 40      R.HEINEN 11-AUG-1976
0000 41
0000 42 Revision history:
0000 43      V03-019 MIRO450      Michael I. Rosenblum      27-Jun-1984
0000 44      Never initiate logins on a terminal marked secure server
0000 45      and noautobaud from unsolicited input.
0000 46
0000 47      V03-018 MIRO370      Michael I. Rosenblum      20-Mar-1984
0000 48      Use TTY$GB_AUTOCHAR to get the character to indicate
0000 49      ready for system password.
0000 50
0000 51      V03-017 MIRO310      Michael I. Rosenblum      07-FEB-1985
0000 52      Fix bug that caused the interrupt key to stop working when
0000 53      in 8 bit mode. Add code to output a different character
0000 54      as a ready for system password character.
0000 55      Fix bug in pasthru mode so it will honor control-s and
0000 56      control-q correctly.
0000 57
```

0000	58	:	V03-016	MIR0082	Michael I. Rosenblum	19-Aug-1983
0000	59	:			Make long word reference to a byte field a byte reference.	
0000	60	:				
0000	61	:	V03-015	MIR0080	Michael I. Rosenblum	14-Jun-1983
0000	62	:			Restructure module.	
0000	63	:				
0000	64	:	V03-014	MIR0051	Michael I. Rosenblum	23-Jun-1983
0000	65	:			Optimize normal character input code path, make jobcontroller	
0000	66	:			notification logic a subroutine, output bell upon notification	
0000	67	:			of job controller on login. Make buffering characters in the	
0000	68	:			typeahead buffer a subroutine.	
0000	69	:				
0000	70	:	V03-013	MIR1050	Michael I. Rosenblum	23-May-1983
0000	71	:			finish removing code for broadcast.	
0000	72	:				
0000	73	:	V03-012	JLV0255	Jake VanNoy	23-MAY-1983
0000	74	:			Add code to allow out-of-band aborts. Set up multi-echo	
0000	75	:			strings to be table driven.	
0000	76	:				
0000	77	:	V03-011	MIR0041	Michael I. Rosenblum	29-Apr-1983
0000	78	:			Cause autobaud to require a readable CR before initiating	
0000	79	:			loginout. Clear the passall optimization bit when EOL is	
0000	80	:			set. Check int before jumping into the getnextchar code path	
0000	81	:			when EOL is set by the passall code path.	
0000	82	:				
0000	83	:	V03-010	MIR0034	Michael I. Rosenblum	07-Apr-1983
0000	84	:			allow LK201 function key F6 to translate to ^C	
0000	85	:				
0000	86	:	V03-009	MIR0032	Michael I. Rosenblum	05-Apr-1983
0000	87	:			Allow control-C,Y and O to echo dec crt strings.	
0000	88	:				
0000	89	:	V03-008	RKS0008	RICK SPITZ	14-MAR-1983
0000	90	:			ADD SUPPORT FOR LOGICAL UCB	
0000	91	:				
0000	92	:	V03-007	MIR0023	Michael I. Rosenblum	24-Jan-1983
0000	93	:			Move the location of setting MULTI in the control-C and Y	
0000	94	:			logic to allow MULTI to be cleared during READONE.	
0000	95	:			Clearing MULTI will stop the Read buffer from	
0000	96	:			being modified after readone deallocates it.	
0000	97	:				
0000	98	:	V03-006	MIR0017	Michael I. Rosenblum	05-Jan-1983
0000	99	:			Change return status of TTY\$PUTNEXTCHAR to include a byte value	
0000	100	:			in the UCB, this will move the information from the condition	
0000	101	:			code bits.	
0000	102	:				
0000	103	:	V03-005	MIR0015	Michael I. Rosenblum	20-Dec-1982
0000	104	:			Change TTY\$V_ST_UNSQL and TTY\$V_ST_GETAHD to TTY\$V_FD_UNSQL	
0000	105	:			and TTY\$V_FD_GETAHD, to reflect changes in the fork dispatcher	
0000	106	:			Change calls to port driver to call the class driver jacket	
0000	107	:			routines.	
0000	108	:				
0000	109	:	V03-004	MIR0014	Michael I. Rosenblum	17-Dec-1982
0000	110	:			Change PORT_XON and PORT_XOFF to CLASS_XON and CLASS_XOFF	
0000	111	:				
0000	112	:	V03-003	MIR0013	Michael I. Rosenblum	16-Dec-1982
0000	113	:			Fix up references to new ucb structure	
0000	114	:				

0000	115	:	V03-002	MIR0011	Michael I. Rosenblum	18-Nov-1982
0000	116	:			Change multiecho to always take a length count and address	
0000	117	:			of a string.	
0000	118	:				
0000	119	:	V03-002	MIR0010	Michael I. Rosenblum	09-Nov-1982
0000	120	:			Move the address of the terminator mask, and the length	
0000	121	:			of the prompt string from the IRP into the terminal read	
0000	122	:			packet. Also move the count of the characters in the	
0000	123	:			buffer from the UCB into the terminal typeahead buffer packet.	
0000	124	:			Restructured typeahead buffer alarm size calculation slightly	
0000	125	:			to use the count from the typeahead buffer packet.	
0000	126	:				
0000	127	:	V03-001	RKS0001	RICK SPITZ	23-SEP-1982
0000	128	:			RESET ALTLEN VALUE WHEN STARTING MULTIECHO STRING, TO INSURE	
0000	129	:			THAT ^Y ECHO DURING READ VERIFY FUNCTIONS PROPERLY.	
0000	130	:				
0000	131	:	V02-026	RKS0026	RICK SPITZ	25-JAN-1982
0000	132	:			CHANGE CONTROL O LOGIC TO WORK WITH BURST MODE OUTPUT	
0000	133	:			(PREVENT WRAP IN MID LINE)	
0000	134	:				
0000	135	:	V02-025	RKS0025	RICK SPITZ	15-DEC-1981
0000	136	:			CHANGE INSPPOST CALL TO WRITE POST IN CTRLC,Y LOGIC.	
0000	137	:				
0000	138	:	V02-024	RKS0024	RICK SPITZ	20-NOV-1981
0000	139	:			ADD OUT OF BAND SUPPORT	
0000	140	:				
0000	141	:	V02-023	JLV0099	Jake VanNoy	27-Oct-1981
0000	142	:			Changed TTYDEFS to \$TTYDEFS.	
0000	143	:				
0000	144	:	V02-022	RKS022	RICK SPITZ	20-AUG-1981
0000	145	:			ADD ALTERNATE TYPEAHEAD SIZE SUPPORT	
0000	146	:				
0000	147	:	V02-021	RKS021	RICK SPITZ	12-AUG-1981
0000	148	:			CORRECT DEFINITION NAMES	
0000	149	:				
0000	150	:	V02-020	JLV0062	Jake VanNoy	10-Aug-1981
0000	151	:			Added autobaud code.	
0000	152	:				
0000	153	:	V02-019	RKS019	RICK SPITZ	27-JULY-1981
0000	154	:			SEVERAL NEW FEATURES HAVE BEEN ADDED TO SUPPORT THE	
0000	155	:			CLASS/PORT STRUCTURE OF THE TERMINAL SERVICES. THESE	
0000	156	:			INCLUDE ENHANCEMENTS TO SUPPORT QUADWORD STATE AND	
0000	157	:			MOVING MOST XON/XOFF LOGIC TO THE PORT DRIVER. THE	
0000	158	:			PORT FUNCTIONS RESUME AND STOP(2) ARE USED TO HANDLE	
0000	159	:			RECEIVED CONTROL S AND Q. LOGIC TO HANDLE CONTROL S	
0000	160	:			DURING A BROADCAST IS ALSO INCLUDED.	
0000	161	:				
0000	162	:	V02-018	RKS018	RICK SPITZ	26-FEB-1981
0000	163	:			REMOVE V2.0 AUDIT TRAILS	
0000	164	:				
0000	165	:				

```
0000 167      .SBTTL Declarations
0000 168
0000 169      :
0000 170      : EXTERNAL SYMBOLS
0000 171      :
0000 172      $IODEF      : DEFINE I/O FUNCTION CODES
0000 173      $IPLDEF     : DEFINE IPL'S
0000 174      $IRPDEF     : DEFINE IRP
0000 175      $PRDEF      : DEFINE PROCESSOR REGISTERS
0000 176      $$$DEF      : DEFINE SYSTEM SERVICE STATUS CODES
0000 177      $UCBDEF     : DEFINE UCB
0000 178      $TASTDEF     : DEFINE OUT OF BAND FLAGS
0000 179      $TTDEF      : DEFINE TERMINAL CHARACTERISTICS
0000 180      $TT2DEF     : DEFINE TERMINAL CHARACTERISTICS
0000 181      $TTYDEF     : DEFINE TERMINAL DRIVER SYMBOLS
0000 182      $TTYMACS    : DEFINE TERMINAL MACROS
0000 183      $TTYDEFS    : DEFINE TERMINAL DEFINITIONS
0000 184
00000000 185      .PSECT $$$115_DRIVER, LONG      : DEFINE NON-PAGED PSECT
```

```
0000 187 .SBTTL TTY$PUTNEXTCHAR - BUFFER CHARACTER
0000 188
0000 189 :++
0000 190 : TTY$PUTNEXTCHAR - BUFFER CHARACTER
0000 191 :
0000 192 : FUNCTIONAL DESCRIPTION:
0000 193 :
0000 194 : THIS ROUTINE IS CALLED BY PORT DRIVERS TO PASS INPUT CHARACTERS.
0000 195 :
0000 196 : CHARACTERS RECEIVED ON NON PASSALL UNITS ARE FILTERED FOR IMMEDIATE
0000 197 : CONTROL SEQUENCES. THESE SEQUENCES REPRESENT:
0000 198 :
0000 199 :     CONTROL Y --    CAUSES THE TYPEAHEAD BUFFER TO BE PURGED, THE
0000 200 :                     ENABLED PROCESS TO RECEIVE AN AST, A "AY" TO
0000 201 :                     BE OUTPUT AND THE CURRENT OPERATION IF ANY TO
0000 202 :                     BE COMPLETED WITH A ZERO TRANSFER COUNT FOR READ
0000 203 :                     AND AS IF CONTROL O FOR WRITE.
0000 204 :
0000 205 :     CONTROL C --    CAUSES THE RECEIVER OF CONTROL C AST'S OR THE RECEIVER
0000 206 :                     OF CONTROL Y AST'S TO BE SIGNALLED AS IN CONTROL Y.
0000 207 :
0000 208 :     CONTROL X --    CAUSES THE CONTENTS OF THE TYPEAHEAD BUFFER
0000 209 :                     TO BE PURGED AND A CONTROL U TO BE INSERTED
0000 210 :                     IN THE INPUT STREAM IF A READ IS IN PROGRESS.
0000 211 :
0000 212 :     CONTROL S --    CAUSES ALL OUTPUT ON UNIT TO STOP UNTIL
0000 213 :                     CONTROL Q,Y OR C.
0000 214 :
0000 215 :     CONTROL Q --    RESETS CONTROL S MODE AND STARTS OUTPUT UP.
0000 216 :
0000 217 : SLAVE MODE ( NO UNSOLICITED INPUT ) UNITS MUST HAVE OUTSTANDING
0000 218 : READS OTHERWISE THE CHARACTER, AFTER CONTROL CHARACTER FILTERING
0000 219 : IS IGNORED.
0000 220 :
0000 221 : INPUTS:
0000 222 :     R3 = CHARACTER TO BUFFER
0000 223 :     R5 = UCB
0000 224 :     R1,R2,R4 ARE AVAILABLE FOR USE
0000 225 :
0000 226 : OUTPUTS:
0000 227 :
0000 228 :     R3 =      0      AND CC = ZERO
0000 229 :           CHAR      AND CC = PLUS
0000 230 :           ADDRESS   AND CC = NEGATIVE
0000 231 :
0000 232 :     R5 = UCB ADDRESS
0000 233 : --
0000 234 :
0000 235 : LONG REACH TABLE
0000 236 :
0000 237 : TAB_CHECKPRE:      BRW    CHECKPRE
0000 238 : TAB_CONTINT:       BRW    CONTINT
0000 239 : TAB_EIGHTBIT:      BRW    EIGHTBIT
0000 240 : TAB_OUTBAND_CHAR:  BRW    OUTBAND_CHAR
0000 241 :
0000 242 : MAIN LINE
0000 243 :
```

014A 31 0000 237 TAB_CHECKPRE: BRW CHECKPRE
01B9 31 0003 238 TAB_CONTINT: BRW CONTINT
01A2 31 0006 239 TAB_EIGHTBIT: BRW EIGHTBIT
020A 31 0009 240 TAB_OUTBAND_CHAR: BRW OUTBAND_CHAR

52	00B8	C5	9E	000C	244	TTY\$PUTNEXTCHAR::	
44	A5	8000	8F	B3	0011	245	MOVAB UCBSQ TT STATE(R5),R2 ; ADDRESS STATUS OF UNIT
		ED	12	0017	246	246	BITW #TT\$M-EIGHTBIT,UCBSL_DEVDEPEND(R5); 8BIT TERMINAL?
53	80	8F	8A	0019	247	247	BNEQ TAB_EIGHTBIT ; IF NEQ THEN YES
				001D	248	248	BICB #^X080,R3 ; STRIP 8TH BIT
				0021	249	249	IF_STATE PASALL,TAB_CHECKPRE ; IF PASSALL THEN OPTIMIZE
51	0105	C5	9A	0021	250	250	
		DB	12	0026	251	251	MOVZBL UCBSB TT INTCONT(R5),R1 ; ARE WE IN THE MIDDLE OF THE
				0028	252	252	BNEQ TAB_CONTINT ; THE INTERRUPT KEY - YES THEN HANDLE IT
				0028	253	253	;
				0028	254	254	;; CHECK FOR OUT OF BAND CHARACTER
				0028	255	255	;;
20	53	91	0028	256	256	256	CMPB R3,#^A/ / ; CONTROL CHARACTER?
	DC	1F	002B	257	257	257	BLSSU TAB_OUTBAND_CHAR ; YES THEN PROCESS IT
			002D	258	258	258	BRB BUFFER_CHAR ; FALL INTO BUFFER CHARACTER

```
002D 260 .SBTTL BUFFER_CHAR - puts character into typeahead buffer
002D 261
002D 262 :++
002D 263 : BUFFER_CHAR - INSERT CHARACTER INTO TYPEAHEAD BUFFER
002D 264 :
002D 265 : FUNCTIONAL DESCRIPTION:
002D 266 :
002D 267 : AT THIS POINT WE KNOW THAT THE CHARACTER DOES NOT REPRESENT SOME IMMEDIATE
002D 268 : ACTION.
002D 269 :
002D 270 : INPUTS:
002D 271 :
002D 272 : R2 = ADDRESS OF THE UNIT STATE VECTOR
002D 273 : R3 = CHARACTER TO BUFFER
002D 274 : R5 = UCB ADDRESS
002D 275 :
002D 276 : OUTPUTS:
002D 277 :
002D 278 : R2,R3,R5 ARE PRESERVED
002D 279 :--
002D 280 : .ENABLE LSB
002D 281 BUFFER_CHAR: ; BUFFER CHARACTER
002D 282 IF NOT STATE READ,40$ ; IF READ THEN BUFFER CHARACTER
0031 283 IF STATE PRE,30$ ; A NOFILTER READ THEN OPTIMIZE
003A 284 BBS #UCB$V_TT_TIMO,UCB$W_DEVSTS(R5),20$; BR IF TIMEOUT
OF 68 A5 01 E0 0035 285 10$:
00B2 30 003A 286 BSBW BUFFER_INSERT ; INSERT THE CHARACTER IN THE BUFFER
003D 287
003D 288 ; BEGIN ECHO ON READ
003D 289
003D 290 BEGIN_ECHO:
003D 291
003D 292 ; START UP OUTPUT IF NOT ALREADY STARTED
003D 293
003D 294 BITW #UCB$M_INT,UCB$W_STS(R5); INTERRUPT EXPECTED?
64 A5 02 B3 003D 295 BNEQ 16$ ; IF NEQ THEN YES
03 12 0041 296 BRW TTY$GETNEXTCHAR ; FIND THE NEXT CHARACTER FOR THIS UNIT
FFBA' 31 0043 297 16$: BRW DISMISS ; EXIT
00A1 31 0046 298
0049 299 ; TIMEOUT ACTIVE - RESET THE TIMER THEN FALL INTO THE NORMAL PATH
0049 300
0049 301 20$:
54 78 A5 D0 0049 302 MOVL UCB$L_SVAPTE(R5),R4 ; ADDRESS READ BUFFER BLOCK
54 36 A4 3C 004D 303 MOVZWL TTY$W_RB_TIMOS(R4),R4 ; GET THE NUMBER OF SECONDS
E7 13 0051 304 BEQL 10$ ; BR IF READ WITH ZERO SECOND TIMEOUT.
00B0 C5 00000000'GF 54 C1 0053 305 ADDL3 R4,G^EXE$GL_ABSTIM,UCB$L_TT_RDUE(R5); RESET THE TIME
DB 11 005D 306 BRB 10$ ; RETURN TO THE MAIN LINE
00FE 31 005F 307 30$: BRW GOPASS
0062 308
0062 309
0062 310 ; NO READ IS CURRENTLY ACTIVE
0062 311
0062 312 40$: BBS #TTY$V_NOTYPEAHD,UCB$L_DEVDEPEND(R5),5$; IF SLAVED TERMINAL, IGNORE C
51 00C0 C5 D0 0067 313 MOVL UCB$L_TT_LOGUCB(R5),RT ; GET LOGICAL UCB ADDRESS
5C A1 B5 006C 314 TSTW UCB$W_REFC(R1) ; UNIT REF COUNT 0?
18 13 006F 315 BEQL 50$ ; IF EQL THEN JOB CONTROLLER POSSIBILITY
60 A1 D5 0071 316 TSTL UCB$L_AMB(R1) ; USER ASSOCIATED MAILBOX?
```

```

      C4 13 0074 317      BEQL 10$      ; IF EQL THEN NO
BF 44 A5 10 E0 0076 318      BBS      #TT$V_MBXDSABL,UCB$L_DEVDEPN2(R5),10$; BR IF NOT ENABLED
BA 68 A5 02 E0 007B 319      BBS      #UCB$V_TT_NOTIF,UCB$Q_DEVSTS(R5),10$; BR IF ALREADY NOTIFIED
      FF7D' 30 0080 320      BSBW      TTYSNOTIFY
      FFB4 31 0083 321 4$:      BRW 10$
      0061 31 0086 322 5$:      BRW DISMISS      ; CONTINUE
      0089 323
      0089 324 ; Before checking terminator, check for autobaud detect
      0089 325
      08 48 A5 01 E1 0089 326 50$:      BBC      #TT2$V_AUTOBAUD,UCB$L_DEVDEPN2(R5),60$ ; Branch if no autobaud
      53 0D 91 008E 327      CMPB      #TTYS$C_CR,R3      ; CR MEANS THAT WE ARE CORRECT
      08 13 0091 328      BEQL 65$      ; SO FALL THRU
      FF6A' 30 0093 329      BSBW      TTYSAUTOBAUD      ; Check for correct baud rate
      E8 48 A5 10 E0 0096 330 60$:      BBS      #TT2$V_SECURE,UCB$L_DEVDEPN2(R5),4$; DON'T EVER INITIATE
      009B 331      ; HERE IF SECURE SERVER IS SET
      E2 0000'CF 53 E1 009B 332 65$:      BBC      R3,W^TTYS$A_STANDARD,4$ ; IF NOT TERMINATOR THEN NOT FOR JOBCTLR
      00000068'GF B5 00A1 333      TSTW      G^SYS$GL_JOBCTLMB+UCB$W_DEVSTS; TERMINALS ENABLED FOR JOBCTLR?
      DD 15 00A7 334      BLEQ 5$      ; IF LEQ THEN DISMISS
      FF54' 30 00A9 335 70$:      BSBW      TTYSNOTIFY      ; NOTIFY THE JOB CONTROLLER
      004D 30 00AC 336      BSBW      BUFFER_INSERT      ; BUFFER THE CHARACTER
      36 64 A5 01 E0 00AF 337      BBS      #UCB$V_INT,UCB$L_STS(R5),DISMISS; DON'T RETURN DATA IF INT EXPECTED
      53 00000000'GF 9A 00B4 338      MOVZBL G^TTYS$B_AUTOCHAR,R3 ; AND RETURN THE AUTOBAUDED CHARACTER
      2D 13 00BB 339      BEQL DISMISS ; NO CHARACTER TO RETURN THEN EXIT
      00BD 340      TIMSET #1,R1,LOCKOUTPUT ; LOCK THE OUTPUT STREAM AND TIME OUT THE CH
      010B C5 01 90 00DE 341      MOVB #1,UCB$B_TT_OUTTYPE(R5) ; SETUP THAT WE RETURNED DATA
      05 00E3 342      RSB ; THEN RETURN
      00E4 343
      00E4 344 ; NO TYPEAHEAD BUFFER - ALLOCATE ONE
      00E4 345
      00E4 346 NO_BUFFER:
      54 01 D0 00E4 347      MOVL #TTYS$V_FD_GETAHD,R4 ; ASK FOR TYPEAHEAD FORK
      FF16' 30 00E7 348      BSBW TTYS$CRE_FORK ; CREATE THE FORK FOR ALLOCATION
      00EA 349
      00EA 350 ; DISMISS INTERRUPT - NOTHING TO DO
      00EA 351
      00EA 352 DISMISS:
      010B C5 94 00EA 353      CLRB UCB$B_TT_OUTTYPE(R5) ; SET NO RETURN CHARACTER
      05 00EE 354      RSB ; RETURN
      00EF 355      .DISABLE LSB
```



```
.SBTTL BUFFER_INSERT - INSERT CHARACTERS INTO THE TYPEAHEAD BUFFER
++
BUFFER_INSERT - BUFFER CHARACTER IN CIRCULAR TYPEAHEAD BUFFER
TEST TO SEE IF THE NUMBER OF CHARACTERS IN THE TYPEAHEAD IS CRITICAL
THIS TEST DOES NOT WORK FOR TYPEAHEAD BUFFERS BIGGER THAN 32K BYTES.
INPUTS:
R3 - CHARACTER TO INSERT
R5 - PHYSICAL UCB ADDRESS
OUTPUTS:
NONE
R4,R1 ARE DESTROYED
--
BUFFER_INSERT:
54 00E4 C5 D0 00EF 357 MOVL UCB$L TT TYPAND(R5),R4 : ADDRESS TYPEAHEAD BUFFER
EE 13 00F4 358 BEQL NO BUFFER : IF EQL THEN NONE
07 E1 00F6 359 BBC #TT2$V ALTYPEAHD,- : SKIP IF NORMAL TYPE AHEAD SIZE
1C 48 A5 00F8 360 UCB$L_DEVDEPND2(R5),55$
51 00000000'GF 00000000'GF A3 00FB 361 SUBW3 G^TTYS$GW ALTALARM,G^TTYS$GW ALTYPAND,R1 : WITHIN N CHARS OF TYPAND F
0030 OC A4 01 51 3D 0107 362 ACBW R1,#1,TTYS$W_TA_INAHD(R4),60$ : BRANCH IF NO
51 00000000'GF A0 010E 363 ADDW2 G^TTYS$GW_ALTALARM,R1 : MAX # TYPEAHEAD CHARS
12 11 0115 364 BRB 57$ : JOIN COMMON PATH
0117 365
51 00000000'GF 08 A3 0117 366 55$: SUBW3 #8,G^TTYS$GW_TYPAHDSZ,R1 : WITHIN 8 CHARS OF TYPAND FULL?
0018 OC A4 01 51 3D 011F 367 ACBW R1,#1,TTYS$W_TA_INAHD(R4),60$ : BRANCH IF NO
51 08 A0 0126 368 ADDW2 #8,R1 : MAX # TYPEAHEAD CHARS
FED4' 30 0129 369 57$: BSBW TTYS$XOFF : SEND XOFF
0C A4 51 B1 0130 390 SET STATE <TYPFUL> : SET UP STOP SEQUENCE
08 18 0134 391 CMPB R1,TTYS$W_TA_INAHD(R4) : TYPEAHEAD BUFFER OVERFLOW?
OC A4 B7 0136 392 BGEQ 60$ : IF GEQ THEN NO OVERFLOW - BUFFER CHARACTER
0139 393 DECB TTYS$W_TA_INAHD(R4) : RESET COUNT ON OVERFLOW
05 013D 394 SET STATE <OVRFLD> : INDICATE OVERFLOW CONDITION
013E 395 RSB
013E 396 :
013E 397 : INSERT CHARACTER IN TYPEAHEAD BUFFER
013E 398
00 B4 53 90 013E 399 60$: MOVB R3,TTYS$L_TA_PUT(R4) : INSERT CHARACTER IN BUFFER
05 64 10 A4 F2 0142 400 AOBLSS TTYS$L_TA_END(R4),TTYS$L_TA_PUT(R4),70$: INDEX AND BR IF NOT AROUND
64 0118 C4 9E 0147 401 MOVAB TTYS$L_TA_DATA(R4),TTYS$L_TA_PUT(R4); RESET POINTER
05 014C 402 70$: RSB
```

```
014D 404 .sbttl Put next character service routines
014D 405
014D 406 :: AVOID TYPEAHEAD BUFFER FOR PASSALL MODE
014D 407
014D 408 .ENABLE LSB
014D 409 CHECKPRE:
014D 410 BBC #TT2$V_PASTHRU,UCB$L_DEVDEPND2(R5),200$; IS THIS TRUE PASSALL MODE?
00000000'EF43 12 E1 0152 411 BITB #TTY$M_CH_CTRL,TTY$A_TYPE[R3]; IS THIS A CONTROL CHAR
20 93 015A 412 BNEQ 250$ ; YES THEN CHECK FOR CONTROL-S OR Q
1E 12 015C 413 200$: IF_NOT_STATE PRE,220$ ; NO READ WAITING THEN BUFFER THE CHARACTER
0160 414
0160 415 :: Pass all optimization verified continue
0160 416
0160 417 GOPASS:
0160 418 MOVL UCB$L_SVAPTE(R5),R4 ; ADDRESS READ BUFFER BLOCK
30 54 78 A5 D0 0164 419 BBS #UCB$V TT_TIMO,UCB$W_DEVSTS(R5),270$; BR IF NO TIMEOUT
68 68 A5 01 0164 419 BBS #UCB$V TT_TIMO,UCB$W_DEVSTS(R5),270$; BR IF NO TIMEOUT
FE94' 30 0169 420 210$: BSBW PASSALL ; OPTIMIZE BY SKIPPING THE TYPEAHEAD
FF77 31 016C 421 IF STATE EOL,230$ ; If read complete then exit
0170 422 BRQ DISMISS ; NORMAL CHARACTER THEN DISMISS THE INTERRUPT
0173 423
0173 424 :: READ TERMINATED
0173 425
0173 426 230$: CLR_STATE PRE ; EOL THEN CLEAR PRE
FEC3 31 0177 427 BRW BEGIN_ECHO ; AND BEGIN THE EOL SEQUENCE
017A 428
017A 429 :: Flow control allowed and the character could be a control-s or q
017A 430
017A 431 250$: CMPB TTY$A_TYPE[R3],#3!TTY$M_CH_CTRL; IS IT A CONTROL-Q
23 00000000'EF43 91 0182 432 BGTRU 200$ ; NO THEN HANDLE NORMALLY
D8 1A 0184 433 BEQL 260$ ; yes then restore flow
0D 13 0186 434 CMPB TTY$A_TYPE[R3],#2!TTY$M_CH_CTRL; IS IT A CONTROL-S
22 00000000'EF43 91 0186 434 CMPB TTY$A_TYPE[R3],#2!TTY$M_CH_CTRL; IS IT A CONTROL-S
CC 12 018E 435 BNEQ 200$ ; NO THEN EXIT
016A 31 0190 436 BRW CONTROLQ ; else stop output
0176 31 0193 437 260$: BRW CONTROLS
0196 438
0196 439 :: jump to buffer the character
0196 440
0196 441 220$: BRW BUFFER_CHAR
0199 442
0199 443 :: RESET TIMERS AND CONTINUE
0199 444
0199 445 270$: MOVZWL TTY$W_RB_TIMOS(R4),R1 ; GET THE NUMBER OF SECONDS
00B0 C5 00000000'GF 51 C1 019D 446 BEQL 210$ ; BR IF READ WITH ZERO SECOND TIMEOUT.
CA 13 019F 447 ADDL3 R1,G*EXE$GL_ABSTIM,UCB$L_TT_RDUE(R5); RESET THE TIME
BE 11 01A9 448 BRB 210$
01AB 449 .DISABLE LSB
01AB 450 .ENABLE LSB
01AB 451
01AB 452 :: 8 BIT CHARACTER IT MAY BE A CSI
01AB 453
01AB 454
01AB 455 EIGHTBIT:
01AB 456 IF_STATE PASALL,CHECKPRE ; IF WE ARE IN PASSALL MODE THEN
01AF 457 ; DON'T PROCESS CSI
53 98 8F 91 01AF 458 CMPB #TTY$C_CSI,R3 ; NO THEN IS IT A CSI
03 12 01B3 459 BNEQ 305$ ; NO THEN CONTINUE NORMALLY
0174 31 01B5 460 BRW CSI ; ELSE TAKE CSI ACTION.
```

```
51 0105 C5 9A 01B8 461 305$: MOVZBL UCB$B_TT_INTCNT(R5),R1 ; ARE WE IN THE MIDDLE OF THE
20 13 01BD 462 BEQL 310$ ; THE INTERRUPT KEY - YES THEN HANDLE IT
01BF 463
01BF 464
01BF 465 ; CHECK FOR INTERRUPT KEY
01BF 466
01BF 467 CONTINT: CLRB UCB$B_TT_INTCNT(R5) ; ASUME IT IS THE WRONG KEY
0000'CF41 0105 C5 94 01C3 468 CMPB R3,W^INTERRUPT_KEY[R1] ; IS THIS THE NEXT CHARACTER
53 91 01C9 469 BNEQ 310$ ; NO THEN THE STATE IS CORRECT SO EXIT
14 12 01CB 470 ADDB3 #1,R1,UCB$B_TT_INTCNT(R5); INCREMENT THE COUNT
0105 C5 51 01 81 01D1 471 CMPB UCB$B_TT_INTCNT(R5),S^#INTERRUPT_KEY_LEN
00' 0105 C5 91 01D6 472 BNEQ 310$
07 12 01D8 473 CLRB UCB$B_TT_INTCNT(R5) ; COMPLETE ESCAPE SEQUENCE THEN RESET
0105 C5 94 01DC 474 BRW CONTROLC
0062 31 01DF 475 310$: CMPB R3,#^A/ / ; IS THIS CHARACTER OUT OF BAND
20 53 91 01E2 476 BLSSU OUTBAND_CHAR ; YES THEN HANDLE CORRECTLY
32 1F 01E4 477 BRW BUFFER_CHAR ; ELSE BUFFER THE CHARACTER
FE46 31 01E7 478 .DISABLE LSB
01E7 479
01E7 480 ; OUT OF BAND CHARACTER PROCESSING
01E7 481
01E7 482 DELOUTBAND:
54 009C C1 DE 01E7 483 MOVAL UCB$B_TL_BANDQUE(R1),R4 ; ADDRESS OF QUEUE
56 00A4 C1 DD 01EC 484 PUSHL R6 ; SAVE R6
00000000'GF 16 DO 01EE 485 MOVL UCB$B_TL_CTLPID(R1),R6 ; GET THE CONTROLLING PID
56 8ED0 01F3 486 JSB G^COM$DE[CTRLASTP ; DELIVER ANY OUTSTANDING ASTS
01F9 487 POPL R6 ; RESTORE R6
01FC 488
01FC 489 ; CHECK FOR INCLUDE OR ABORT
01FC 490
01FC 491 BBCC #TAST$V_ABO,R3,5$ ; CHECK ABORT FIRST
7C A5 0F 53 OE E5 0200 492 MOVZWL #55$ CONTROLC,UCB$W_BOFF(R5) ; SET STATUS
0651 8F 3C 0206 493 CLRW UCB$B_TT_MULTILEN(R5) ; CLEAR, NO EHCO TO PERFORM
00DC C5 B4 020A 494 BSBW TTY$ABORT_IO ; ABORT I/O
FDF3' 30 020D 495 BRB 8$ ; CONTINUE
04 11 020F 496 5$: BBSC #TAST$V_INC,R3,CHECASE ; DONT STRIP CHARACTER
OE 53 OF E4 0213 497 8$: BRW DISMISS
FED4 31 0216 498
0216 499 ; CHECK FOR CONTROL SEQUENCE CHARACTER
0216 500
0216 501 OUTBAND_CHAR:
0216 502
51 00C0 C5 DO 0216 503 MOVL UCB$B_TT_LOGUCB(R5),R1 ; GET LOGICAL UCB ADDRESS
C6 0098 C1 53 E0 0218 504 BBS R3,UCB$B_TL_OUTBAND(R1),DELOUTBAND; NOT IN SUMMARY MASK
0221 505
0221 506 CHECASE:
0221 507 CASE W^TTY$A_TYPE[R3],TYPE=B,LIMIT=#1@TTY$V_CH_CTRL,-
0221 508 <CONTROLC,CONTROLO,CONTROLQ,CONTROLS,CONTROLX,CONTROLY,ESCAPE,CSI>
FDF2 31 0238 509 BRW BUFFER_CHAR
```


TTYCHARI
V04-000

H 3
- terminal input character routines 16-SEP-1984 02:20:10 VAX/VMS Macro V04-00
PRE-TYPEAHEAD CONTROL CHARACTER ACTION R 5-SEP-1984 04:16:13 [TTDRVR.SRC]TTYCHARI.MAR;1 Page 12
(7)

```
023B 511 .SBTTL PRE-TYPEAHEAD CONTROL CHARACTER ACTION ROUTINES.
023B 512
023B 513 : ENTRY FROM CONTROLY AND CONTROLC
023B 514
023B 515 CANCEL_CTRL:
023B 516 BSBW TTY$RESUME : RESUME ANY PORT OUTPUT
FDC2' 30 023B 516 BRW BEGIN_ECHO : BEGIN OUTPUT
FDFC 31 023E 517
```

```
0241 519
0241 520      .SBTTL CONTROLC, CONTROLY handlers
0241 521
0241 522      .ENABL LSB
0241 523
0241 524      ++
0241 525      CONTROLC = SIGNAL CONTROL C INPUT
0241 526      CONTROLY = SIGNAL CONTROL Y INPUT
0241 527
0241 528      FUNCTIONAL DESCRIPTION:
0241 529
0241 530      THIS ROUTINE IS ENTERED WHEN A CONTROL C OR Y IS TYPED.
0241 531      THE ACTION IS TO SIGNAL, VIA AN AST, THE HOLDER OF THE AST ENABLE.
0241 532      IF NO ENABLE IS PRESENT, AN ATTEMPT IS MADE TO SIGNAL THE JOB CONTROLLER
0241 533      IF IT HAS NOT BEEN SIGNED.
0241 534
0241 535      INPUTS:
0241 536
0241 537      R2 = ADDRESS OF THE UNIT STATE VECTOR
0241 538      R5 = UCB ADDRESS
0241 539
0241 540      OUTPUTS:
0241 541
0241 542      R2 = ADDRESS OF THE UNIT STATE VECTOR
0241 543      R5 = UCB ADDRESS
0241 544
0241 545      --
0241 546
0241 547      M_CTRLC = 1
0241 548      M_REGIS = 2
0241 549      M_DECCRT = 4
0241 550
0241 551      CONTROLC:
0241 552      MOVL      UCB$$_TT_LOGUCB(R5),R1      ; ENTRY FOR CONTROL C
0241 553      MOVAL     UCB$$_TL_CTRLC(R1),R4      ; GET LOGICAL UCB ADDRESS
0241 554      TSTL      (R4)                        ; GET ADDRESS OF CONTROL C AST LIST
0241 555      BEQL      10$                          ; EMPTY?
0241 556      MOVZWL     #SS$_CONTROLC,UCB$$_BOFF(R5) ; IF EQL THEN NO CTRLC
0241 557      MOVZBL     #M_CTRLC,R3                ; SET STATUS AND ZERO COUNT
0241 558      BRB       15$                          ; Set "Cancel"
0241 559      ; Branch
0241 560
0241 561      : CONTROL Y PROCESSING
0241 562
0241 563      CONTROLY:
0241 564      MOVL      UCB$$_TT_LOGUCB(R5),R1      ; GET LOGICAL UCB ADDRESS
0241 565      MOVAL     UCB$$_TL_CTRLY(R1),R4      ; GET ADDRESS OF CONTROL Y AST LIST
0241 566      TSTL      (R4)                        ; EMPTY LIST?
0241 567      BNEQ      13$                          ; No, branch forward.
0241 568      BRW       TAB_BUFFER                  ; Yes. Don't process it.
0241 569
0241 570      13$:
0241 571      MOVZWL     #SS$_CONTROLY,UCB$$_BOFF(R5) ; SET STATUS AND ZERO COUNT
0241 572      CLRL      R3                          ; Set "Interrupt"
0241 573
0241 574      : Common ^C and ^Y code
0241 575
```

00000001
00000002
00000004

51 00C0 C5 D0
54 0094 C1 DE
64 D5
7C A5 0651 8F 3C
53 01 9A
19 11

51 00C0 C5 D0
54 0090 C1 DE
64 D5
03 12
0097 31

7C A5 0611 8F 3C
53 D4

03 48 A5 1D E1	0273 576 15\$:	BBC	#TT2\$V DECCRT,UCB\$L_DEVDEPND2(R5),18\$; IS THIS A DECCRT?
53 04 88	0273 577	BISB	#M_DECCRT,R3 ; YES THEN DO A DECCRTECHO
03 48 A5 19 E1	0278 578	BBC	#TT2\$V REGIS,UCB\$L_DEVDEPND2(R5),20\$; IS THIS REGIS
53 02 88	027B 579 18\$:	BISB	#M_REGIS,R3 ; YES THEN GET OUT OF REGIS MODE
	0280 580		
	0283 581 20\$:		
51 0000'CF D0	0283 582	IF STATE NINTMULTI,25\$; DON'T INTERRUPT THE NON-INTERUPT MULTIECHOS
53 6143 D0	0287 583	MOVL W^TTYSA_INTIECHO,R1	; FETCH TABLE POINTER
00DC C5 83 9B	028C 584	MOVL (R1)[R3],R3	; AND OFFSET
00DB C5 53 D0	0290 585	MOVZBW (R3)+,UCB\$W_TT_MULTILEN(R5);	SETUP THE LENGTH OF THE MULTIECHO STRIN
	0295 586	MOVL R3,UCB\$L_TT_MULTI(R5)	; SET UP FOR MULTIECHO
	029A 587	SET_STATE NINTMULTI	; MAKE THIS ONE OF THE NON-INTERUPTABLE
	029E 588		; MULTIECHO STRING
56 00C0 C5 D0	029E 589 25\$:	PUSHL R6	
56 00A4 C6 D0	02A0 590	MOVL UCB\$L_TT_LOGUCB(R5),R6	; GET THE LOGICAL UCB ADDRESS
00000000'GF 16	02A5 591	MOVL UCB\$L_TL_CTLPID(R6),R6	; THEN GET THE PID
56 8ED0	02AA 592	JSB G^COM\$DECATTNASTP	; DELIVER ATTENTION ASTS FOR THIS PID
	02B0 593	POPL R6	
	02B3 594		
FD4A' 30	02B3 595	BSBW TTY\$ABORT_IO	
FFB2 31	02B6 596	BRW CANCEL_CTRLS	; CANCEL CONTROL S AND BEGIN ECHO


```
02B9 598 .SBTTL CONTROLO handler
02B9 599
02B9 600 :++
02B9 601 : CONTROLO - START OR STOP OUTPUT ON UNIT
02B9 602 :
02B9 603 : FUNCTIONAL DESCRIPTION:
02B9 604 :
02B9 605 : THIS ROUTINE TOGGLES THE OUTPUT ENABLE OF A UNIT.
02B9 606 : OUTPUT IS STOPPED UNTIL THE NEXT READ OPERATION, IOS_WRTCANCTRLO
02B9 607 : OR CONTROL 0.
02B9 608 :
02B9 609 : INPUTS:
02B9 610 :
02B9 611 : R2 = ADDRESS OF THE UNIT STATE VECTOR
02B9 612 : R5 = UCB ADDRESS
02B9 613 :
02B9 614 : OUTPUTS:
02B9 615 :
02B9 616 : R2 = ADDRESS OF THE UNIT STATE VECTOR
02B9 617 : R5 = UCB ADDRESS
02B9 618 :--
02B9 619
00000001 02B9 620 M_CTRLO_DEC = 1
00000002 02B9 621 M_CTRLO_ON = 2
02B9 622
02B9 623 CONTROLO:
02B9 624
02B9 625 IF NOT STATE READ,120$ : IF NOT READ THEN HONOR
02B9 626 IF NOT STATE CTRLR,TAB DISMISS : IF CONTROL R THEN IGNORE
04 A2 01 AC 02C1 627 120$: XORW #<TTY$M ST_CTRLO>,4(R2) : FLOP CONTROL 0 BIT
02C5 628 IF NOT STATE CTRLO,OUTPUTON : IF NOW CLEAR THEN START OUTPUT
FD34' 30 02C9 629 BSBW TTY$ABORT : ABORT PORT OUTPUT
53 D4 02CC 630 CLRL R3 : CLEAR ECHO FLAG
03 11 02CE 631 BRB CTRLO_ECHO : STARTUP THE OUTPUT
02D0 632 :
02D0 633 : RESTART OUTPUT ON STOPPED UNIT
02D0 634 :
02D0 635 OUTPUTON:
53 02 D0 02D0 636 MOVL #M_CTRLO_ON,R3 : SET FLAG
02D3 637
02D3 638 CTRLO_ECHO:
02D3 639 IF STATE NINTMULTI,TAB CANCEL_CTRL$; don't bother non-interruptable multiecho
02D7 640 SET_STATE <MULTI,NINTMULTI>
03 48 A5 E1 02DF 641 BBC #TT2$V DECCRT,-
53 01 A8 02E1 642 UCB$L DEVDEPN2(R5),150$ ; BRANCH IF NOT DECCRT
02E4 643 BISW #M_CTRLO_DEC,R3
02E7 644 150$:
51 0000'CF D0 02E7 645 MOVL W*TTY$A CTRLOECHO,R1 : INSERT ADDRESS OF STRING
53 6143 D0 02EC 646 MOVL (R1)(R3),R3 : AND OFFSET
00DC C5 83 9B 02F0 647 MOVZBW (R3)+,UCB$W TT_MULTILEN(R5); SETUP THE LENGTH OF THE MULTIECHO STRIN
00D8 C5 53 D0 02F5 648 MOVL R3,UCB$L_TT_MULTI(R5) : SET UP FOR MULTIECHO
02FA 649 TAB_CANCEL_CTRL$:
FF3E 31 02FA 650 BRQ CANCEL_CTRL$ : RESUME OUTPUT
```

```
02FD 652 .SBTTL CONTROLQ handler
02FD 653
02FD 654 :++
02FD 655 CONTROLQ - START OUTPUT ON CONTROL S STOPPED UNIT
02FD 656
02FD 657 FUNCTIONAL DESCRIPTION:
02FD 658
02FD 659 THIS ROUTINE STARTS OUTPUT ON A UNIT WHICH IS STOPPED BECAUSE
02FD 660 OF CONTROL S.
02FD 661
02FD 662 INPUTS:
02FD 663
02FD 664 R2 = ADDRESS OF THE UNIT STATE VECTOR
02FD 665 R5 = UCB ADDRESS
02FD 666
02FD 667 OUTPUTS:
02FD 668
02FD 669 R2 = ADDRESS OF THE UNIT STATE VECTOR
02FD 670 R5 = UCB ADDRESS
02FD 671 :--
02FD 672
02FD 673 CONTROLQ:
44 A5 05 E0 02FD 674 BBS #TTSV TTSYNC,UCBSL_DEVDEPEND(R5),-
FB 0301 675 TAB_CANCEL_CTRL ; TERMINAL SYNCH?
0302 676 TAB_BUFFER:
0302 677 IF STATE PRE,TAB GOPASS ; PASSALL OPTOMIZATION
FD24 31 0306 678 BRW BUFFER_CHAR ; BUFFER THE CHARACTER NORMALLY
FE54 31 0309 679 TAB_GOPASS:
0309 680 BRW GOPASS
```

```
030C 682      .SBTTL CONTROLS handler
030C 683
030C 684      :++
030C 685      :CONTROLS - STOP OUTPUT TO UNIT
030C 686
030C 687      :FUNCTIONAL DESCRIPTION:
030C 688
030C 689      :THIS ROUTINE IS ENTERED WHEN THE USER TYPES A CONTROL S.
030C 690      :ON UNITS THAT DO NOT HAVE THE TT$V TT$SYNC CHARACTERISTIC, THE OUTPUT
030C 691      :IS STOPPED BY TURNING THE OUTPUT INTERRUPT ENABLE BIT IN THE UCB
030C 692      :OFF.
030C 693
030C 694      :INPUTS:
030C 695
030C 696      :      R2 = ADDRESS OF THE UNIT STATE VECTOR
030C 697      :      R5 = UCB ADDRESS
030C 698
030C 699      :OUTPUTS:
030C 700
030C 701      :      R2 = ADDRESS OF THE UNIT STATE VECTOR
030C 702      :      R5 = UCB ADDRESS
030C 703      :--
030C 704
030C 705      CONTROLS:
030C 706      BBC      #TT$V TT$SYNC,UCB$$_DEVDEPEND(R5),TAB BUFFER; TERMINAL SYNCH?
0311 707      BSBW      TT$STOP      ; STOP PORT OUTPUT
0314 708      TAB_DISMISS:
0314 709      BRW      DISMISS      ; AND RETURN TO DRIVER CODE
```

F1 44 A5 05 E1
FCEC' 30
FDD3 31


```
0317 711 .SBTTL CONTROLX handler
0317 712
0317 713 :++
0317 714 : CONTROLX - PURGE TYPEAHEAD BECAUSE OF CONTROL X INPUT
0317 715 :
0317 716 : FUNCTIONAL DESCRIPTION:
0317 717 :
0317 718 : THIS ROUTINE PURGES THE TYPEAHEAD BUFFER AND IF THE REASON IS THAT
0317 719 : A CONTROL X WAS TYPED DURING A READ THEN A CONTROL U IS INSERTED IN
0317 720 : THE INPUT STREAM TO PURGE THE CURRENT INPUT LINE.
0317 721 :
0317 722 : INPUTS:
0317 723 :
0317 724 : R2 = ADDRESS OF THE UNIT STATE VECTOR
0317 725 : R5 = UCB ADDRESS
0317 726 :
0317 727 : OUTPUTS:
0317 728 :
0317 729 : R2 = ADDRESS OF THE UNIT STATE VECTOR
0317 730 : R3 = CONTROL U - IF CONTROL X TYPED DURING READ
0317 731 : R5 = UCB ADDRESS
0317 732 :--
0317 733
0317 734 CONTROLX:
FCE6' 30 0317 735 BSBW TTY$PURGE_AHEAD ; PURGE TYPEAHEAD
53 15 9A 031A 736 IF NOT STATE_READ,TAB DISMISS ; DONE IF NO READ IN PROGRESS
FFDE 31 031E 737 MOVZBL #TTY$C_CTRLU,R3 ; FORCE A CONTROL U IN INPUT STREAM
0321 738 BRW TAB_BUFFER ; CONTINUE AND BUFFER CHARACTER
```

```

0324 740 .SBTTL ESCAPE, CSI handler
0324 741 :++
0324 742 : Escape - introducer for the cancel/interrupt key
0324 743 :
0324 744 :--
0324 745 ESCAPE:
0105 C5 01 90 0324 746 MOVW #1,UCBSB TT_INTCNT(R5) ; SET INTERRUPT COUNT
          FD01 31 0329 747 BRW BUFFER_CHAR
032C 748 :++
032C 749 : CSI - CANCEL INTERRUPT INTRODUCER
032C 750 :
032C 751 :--
032C 752 CSI:
0105 C5 02 90 032C 753 MOVW #2,UCBSB TT_INTCNT(R5) ; MOVE BY THE ESC [ IN THE STRING
          FCF9 31 0331 754 BRW BUFFER_CHAR

```

TTYCHARI
V04-000

- terminal input character routines C 4
End of module

16-SEP-1984 02:20:10 VAX/VMS Macro V04-00
5-SEP-1984 04:16:13 [TTDRVR.SRC]TTYCHARI.MAR;1

Page 20
(15)

0334 756 .SBTTL End of module
0334 757
0334 758 .END

TTYCHARI
Symbol table

D 4
- terminal input character routines

16-SEP-1984 02:20:10 VAX/VMS Macro V04-00
5-SEP-1984 04:16:13 [TTDRV.RSRC]TTYCHARI.MAR;1

Page 21
(15)

```
BEGIN ECHO          0000003D R 02
BUFFER_CHAR        0000002D R 02
BUFFER_INSERT      000000EF R 02
CANCEL_CTRL        0000023B R 02
CHECKPRE           00000221 R 02
CHECKPRE           0000014D R 02
CNT                = 00000001
COM$DELATTNASTP    ***** X 02
COM$DELCTR$LASTP   ***** X 02
CONTINT            000001BF R 02
CONTROL           00000241 R 02
CONTROL           000002B9 R 02
CONTROL           000002FD R 02
CONTROLS          0000030C R 02
CONTROLX          00000317 R 02
CONTROLY          0000025A R 02
CSI               0000032C R 02
CTRLO ECHO        000002D3 R 02
DELOUTBAND        000001E7 R 02
DISMISS           000000EA R 02
EIGHTBIT          000001AB R 02
ESCAPE            00000324 R 02
EXESGL_ABSTIM     ***** X 02
F                = 00000000
GOPASS            00000160 R 02
INTERRUPT_KEY     ***** X 02
INTERRUPT_KEY_LEN ***** X 02
M_CTRL           = 00000001
M_CTRL_DEC        = 00000001
M_CTRL_ON         = 00000002
M_DECCRT          = 00000004
M_REGIS           = 00000002
NO_BUFFER         000000E4 R 02
OUTBAND_CHAR      00000216 R 02
OUTPUTOR          000002D0 R 02
PASSALL           ***** X 02
SS$_CONTROL       = 00000651
SS$_CONTROLY      = 00000611
SYS$GL_JOBCTLM    ***** X 02
T                = 0000003B
TAB_BUFFER        00000302 R 02
TAB_CANCEL_CTRL   000002FA R 02
TAB_CHECKPRE      00000000 R 02
TAB_CONTINT       00000003 R 02
TAB_DISMISS       00000314 R 02
TAB_EIGHTBIT      00000006 R 02
TAB_GOPASS        00000309 R 02
TAB_OUTBAND_CHAR  00000009 R 02
TASTSV_ABO        = 0000000E
TASTSV_INC        = 0000000F
TTSV_EIGHTBIT     = 00008000
TTSV_MBXDSABL     = 00000010
TTSV_NOTYPEAHD    = 00000002
TTSV_TTSYN        = 00000005
TT2SV_ALTYPEAHD   = 00000007
TT2SV_AUTOBAUD    = 00000001
TT2SV_DECCRT      = 0000001D
```

```
TT2SV_PASTHRU      = 00000012
TT2SV_REGIS        = 00000019
TT2SV_SECURE       = 00000010
TTSVABORT          ***** X 02
TTSVABORT_10       ***** X 02
TTSV AUTOBAUD      ***** X 02
TTSV_ACTRLOECHO    ***** X 02
TTSV_INTECHO       ***** X 02
TTSV_STANDARD      ***** X 02
TTSV_TYPE          ***** X 02
TTSV CRE FORK      ***** X 02
TTSV CR            = 0000000D
TTSV CSI           = 0000009B
TTSV CTRLU         = 00000015
TTSVGB AUTOCHAR    ***** X 02
TTSVGETNEXTCHAR    ***** X 02
TTSVGTW_ALTALARM   ***** X 02
TTSVGTW_ALTPAHD    ***** X 02
TTSVGTW_TYPAHDSZ   ***** X 02
TTSV TA_DATA       = 00000118
TTSV TA_END        = 00000010
TTSV TA_PUT        = 00000000
TTSV CH_CTRL       = 00000020
TTSV ST_CTRL       = 00000001
TTSV ST_MULTI      = 00000040
TTSV ST_NINTMULTI  = 08000000
TTSV ST_OVRFLO     = 00010000
TTSV ST_PRE        = 04000000
TTSV ST_TYFUL      = 00001000
TTSV NOTIFY        ***** X 02
TTSV PURGE AHEAD   ***** X 02
TTSV PUTNEXTCHAR   0000000C RG 02
TTSV RESUME        ***** X 02
TTSV STOP          ***** X 02
TTSV CH_CTRL       = 00000005
TTSV FD_GETAHD     = 00000001
TTSV PC_NOTIME     = 00000000
TTSV SX_CTRL       = 00000020
TTSV SX_CTRLR      = 00000032
TTSV SX_EOL        = 00000008
TTSV SX_MULTI      = 00000006
TTSV SX_NINTMULTI  = 0000003B
TTSV SX_OVRFLO     = 00000030
TTSV SX_PASALL     = 00000022
TTSV SX_PRE        = 0000003A
TTSV SX_READ       = 0000000C
TTSV SX_TYFUL      = 0000002C
TTSV RB_TIMOS      = 00000036
TTSV TA_INAHD      = 0000000C
TTSV XOFF          ***** X 02
UCBSB TT_INTCT     = 00000105
UCBSB TT_OUTTYPE   = 0000010B
UCBSL_AMB          = 00000060
UCBSL_DEVDEPEND    = 00000044
UCBSL_DEVDEPN2     = 00000048
UCBSL DUETIM       = 0000006C
UCBSL STS          = 00000064
```

TT
VO

TTYCHARI
Symbol table

- terminal input character routines E 4

16-SEP-1984 02:20:10 VAX/VMS Macro V04-00
5-SEP-1984 04:16:13 [TTDRVR.SRC]TTYCHARI.MAR;1

Page 22
(15)

UCBSL_SVAPTE = 00000078
UCBSL_TL_BANDQUE = 0000009C
UCBSL_TL_CTLPID = 000000A4
UCBSL_TL_CTRLC = 00000094
UCBSL_TL_CTRLY = 00000090
UCBSL_TL_OUTBAND = 00000098
UCBSL_TT_LOGUCB = 000000C0
UCBSL_TT_MULTT = 000000D8
UCBSL_TT_RDUE = 000000B0
UCBSL_TT_TYPAHD = 000000E4
UCBSM_INT = 00000002
UCBSM_TIM = 00000001
UCBSQ_TT_STATE = 000000B8
UCBSV_INT = 00000001
UCBSV_TT_NOTIF = 00000002
UCBSV_TT_TIMO = 00000001
UCBSW_BOFF = 0000007C
UCBSW_DEVSTS = 00000068
UCBSW_REFC = 0000005C
UCBSW_STS = 00000064
UCBSW_TT_MULTILEN = 000000DC
UCBSW_TT_PRTCTL = 00000122
W0 = 00000040
W1 = 00000008
X = 00000000
X0 = 00000000
X1 = 00000003
Z0 = 00000000
Z1 = 00000003

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$115_DRIVER	00000334 (820.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.95
Command processing	118	00:00:00.41	00:00:02.07
Pass 1	527	00:00:14.67	00:00:54.15
Symbol table sort	0	00:00:02.44	00:00:09.69
Pass 2	143	00:00:02.81	00:00:10.12
Symbol table output	17	00:00:00.11	00:00:00.36
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	838	00:00:20.51	00:01:18.36

The working set limit was 1800 pages.

120967 bytes (237 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2253 non-local and 38 local symbols.
758 source lines were read in Pass 1, producing 15 object records in Pass 2.
48 pages of virtual memory were used to define 45 macros.

-----+
! Macro library statistics !
-----+-----+

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	17
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	25

2589 GETS were required to define 25 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$;TTYCHARI/OBJ=OBJ\$;TTYCHARI MSRC\$;TTYCHARI/UPDATE=(ENH\$;TTYCHARI)+EXECMLS/LIB

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY